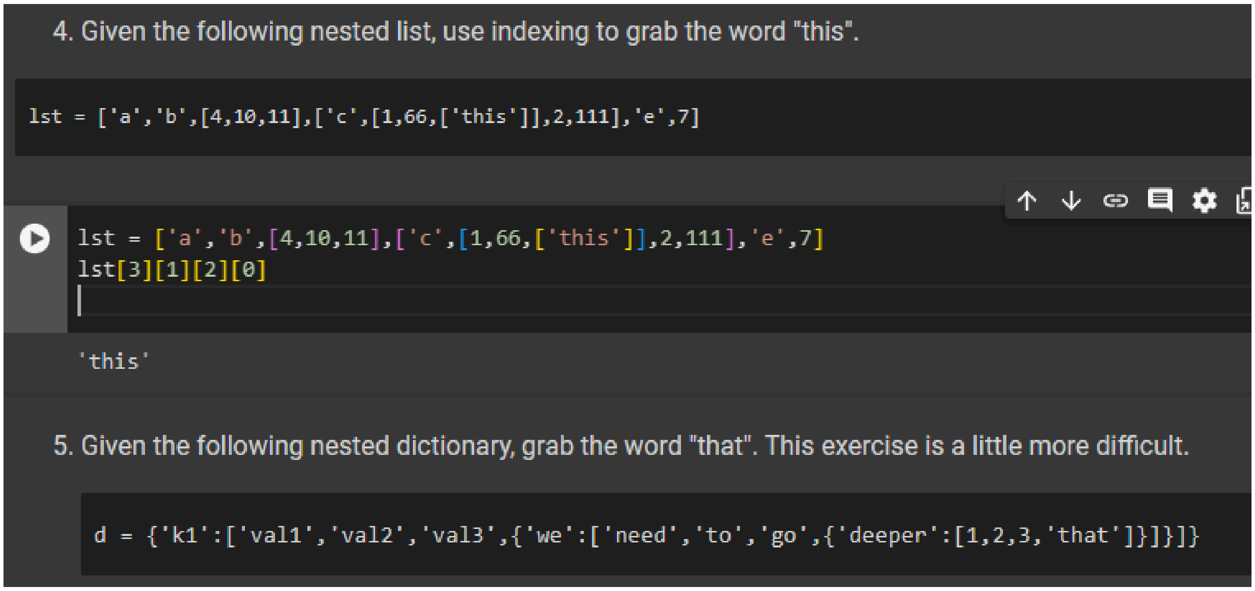
I am Tristan Neal U. Santos, Student of CPE32S9 and this document contain the list of parts I google search to continue the progress of this activity.

1. I searched an example about Given the following nested list, use indexing but the complicated part that lets you detect the word ‘that’.



4. Given the following nested list, use indexing to grab the word "this".

```
lst = ['a','b',[4,10,11],['c',[1,66,['this']],2,111],'e',7]
```

```
lst = ['a','b',[4,10,11],['c',[1,66,['this']],2,111],'e',7]
lst[3][1][2][0]
```

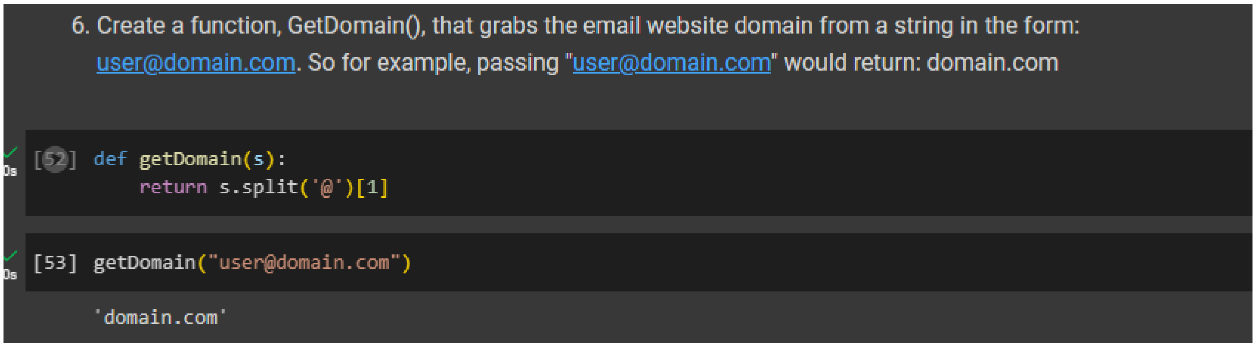
```
'this'
```

5. Given the following nested dictionary, grab the word "that". This exercise is a little more difficult.

```
d = {'k1':['val1','val2','val3',{'we':['need','to','go',{'deeper':[1,2,3,'that']}]}]}
```

Reference:
https://www.tomasbeuzen.com/python-programming-for-data-science/practice-exercises/chapter1-basics-practice.html

2. I searched about “Create a function, grab the email website domain from a string in the form.”



6. Create a function, GetDomain(), that grabs the email website domain from a string in the form: user@domain.com. So for example, passing "user@domain.com" would return: domain.com

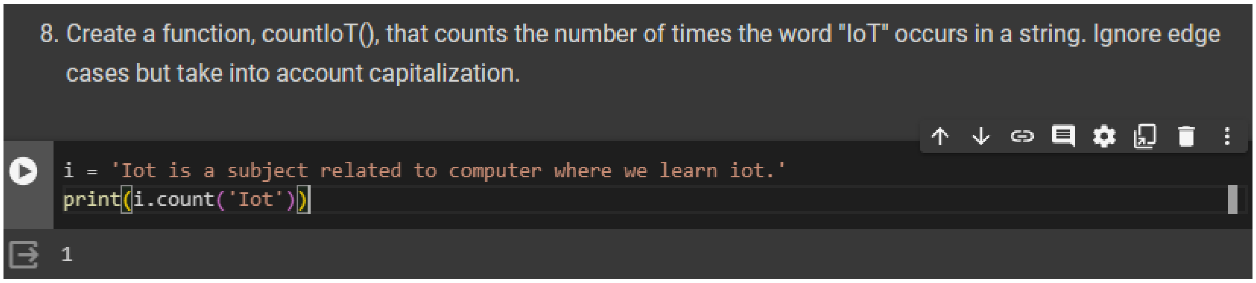
```
[52] def getDomain(s):
         return s.split('@')[1]
```

```
[53] getDomain("user@domain.com")
```

```
'domain.com'
```

https://www.studocu.com/row/messages/question/4547693/c-create-a-function-that-grabs-the-email-website-domain-from-a-string-in-the-form

3. I searched for an example that counts the number of times the word is mentioned.”



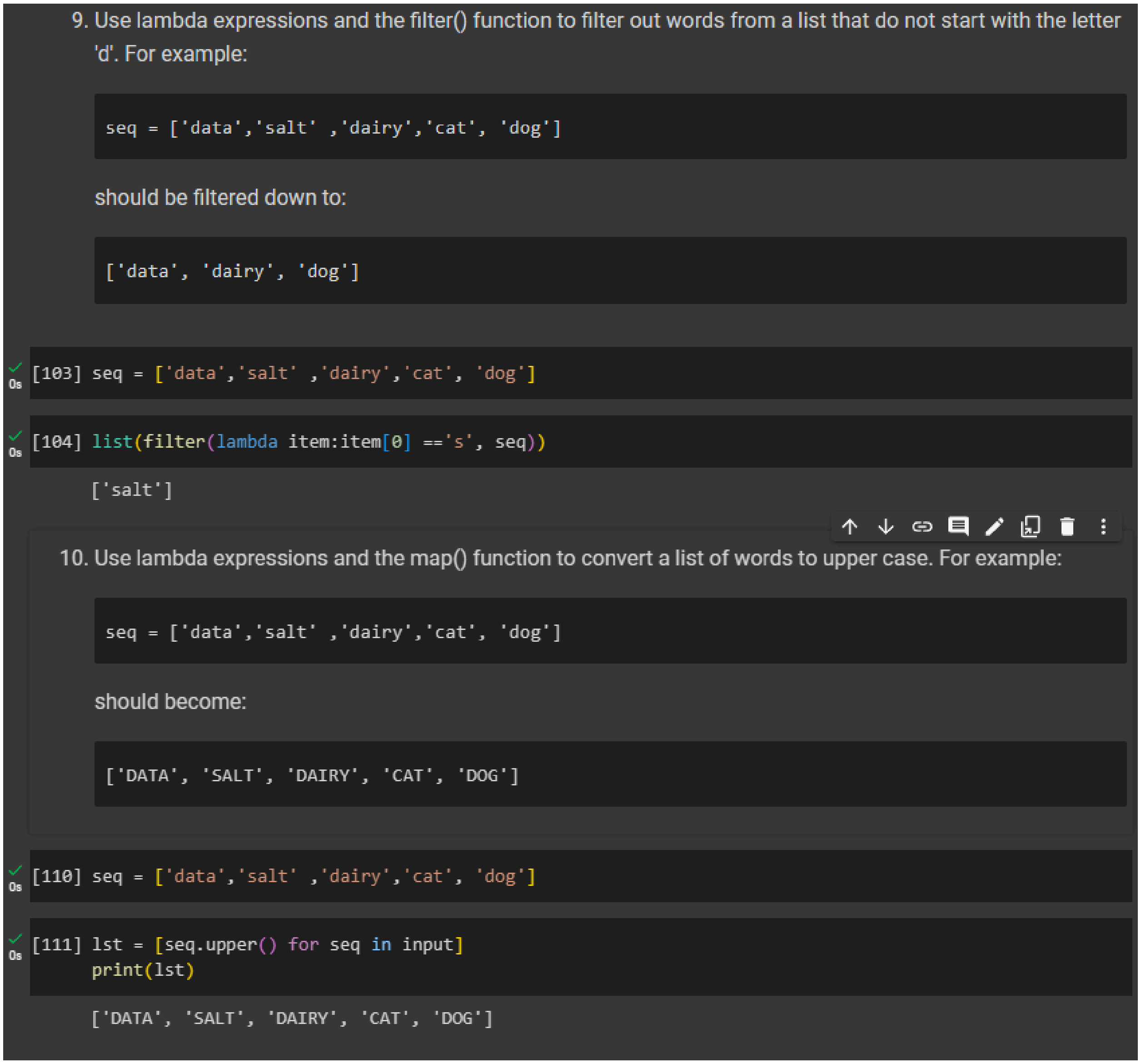
8. Create a function, countIoT(), that counts the number of times the word "IoT" occurs in a string. Ignore edge cases but take into account capitalization.

```
i = 'Iot is a subject related to computer where we learn iot.'
print(i.count('Iot'))
```

```
1
```

Reference:
https://note.nkmk.me/en/python-str-count/#:~:text=a%20substring%20position)-,Count%20characters%20and%20substrings%20in%20a%20string%3A%20count(),or%20substrings%20in%20a%20string.&text=If%20the%20second%20argument%20start,slice%20%5Bstart%3Aend%5D%20.

4. I searched for an example about lambda expressions.



9. Use lambda expressions and the filter() function to filter out words from a list that do not start with the letter 'd'. For example:

```
seq = ['data','salt' ,'dairy','cat', 'dog']
```

should be filtered down to:

```
['data', 'dairy', 'dog']
```

```
[103] seq = ['data','salt' ,'dairy','cat', 'dog']
```

```
[104] list(filter(lambda item:item[0] =='s', seq))
```

```
['salt']
```

10. Use lambda expressions and the map() function to convert a list of words to upper case. For example:

```
seq = ['data','salt' ,'dairy','cat', 'dog']
```

should become:

```
['DATA', 'SALT', 'DAIRY', 'CAT', 'DOG']
```

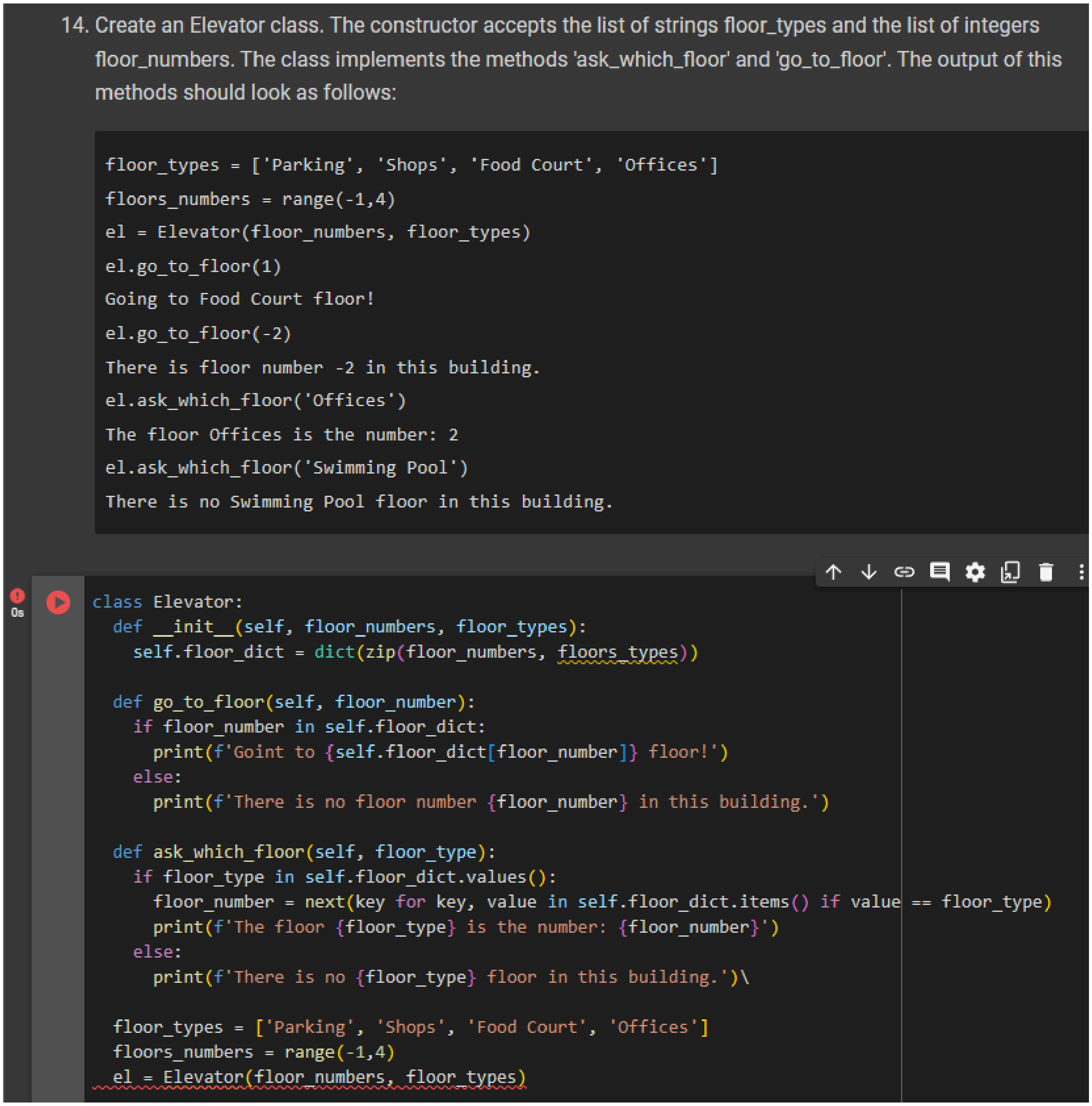
```
[110] seq = ['data','salt' ,'dairy','cat', 'dog']
```

```
[111] lst = [seq.upper() for seq in input]
      print(lst)
```

```
['DATA', 'SALT', 'DAIRY', 'CAT', 'DOG']
```

Reference:
https://www.geeksforgeeks.org/python-convert-case-of-elements-in-a-list-of-strings/

5. I searched about examples of class



14. Create an Elevator class. The constructor accepts the list of strings floor_types and the list of integers floor_numbers. The class implements the methods 'ask_which_floor' and 'go_to_floor'. The output of this methods should look as follows:

```
floor_types = ['Parking', 'Shops', 'Food Court', 'Offices']
floors_numbers = range(-1,4)
el = Elevator(floor_numbers, floor_types)
el.go_to_floor(1)
Going to Food Court floor!
el.go_to_floor(-2)
There is floor number -2 in this building.
el.ask_which_floor('Offices')
The floor Offices is the number: 2
el.ask_which_floor('Swimming Pool')
There is no Swimming Pool floor in this building.
```

```
class Elevator:
  def __init__(self, floor_numbers, floor_types):
    self.floor_dict = dict(zip(floor_numbers, floors_types))

  def go_to_floor(self, floor_number):
    if floor_number in self.floor_dict:
      print(f'Goint to {self.floor_dict[floor_number]} floor!')
    else:
      print(f'There is no floor number {floor_number} in this building.')

  def ask_which_floor(self, floor_type):
    if floor_type in self.floor_dict.values():
      floor_number = next(key for key, value in self.floor_dict.items() if value == floor_type)
      print(f'The floor {floor_type} is the number: {floor_number}')
    else:
      print(f'There is no {floor_type} floor in this building.')\

  floor_types = ['Parking', 'Shops', 'Food Court', 'Offices']
  floors_numbers = range(-1,4)
  el = Elevator(floor_numbers, floor_types)
```